MS Appeal Brief-Patents
PATENT
0579-1097

IN THE U.S. PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of	Appeal No.
Jean-Bernard FISCHER et al.	Conf. 5286
Application No. 10/540,219	Group 2431
Filed January 17, 2006	Examiner M. Vaughan

METHOD AND DEVICE FOR MAKING SECURE EXECUTION OF A COMPUTER PROGRAMME

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Assistant Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

MAY IT PLEASE YOUR HONORS:

(i) **Real Party in Interest**

The real party of interest in this appeal is Oberthur Card Systems SA, 102, Boulevard Malesherbes 75017 Paris, FRANCE.

(ii) **Related Appeals and Interferences**

None.

(iii) **Status of Claims**

Claims 1-9, 11-21, 23-30, 32, 33, 35-40 are pending and stand rejected. This appeal is taken from the final rejection of claims 1-9, 11-21, 23-30, 32, 33, 35-40.

(iv) **Status of Amendments**

All amendments have been entered.

(v) **Summary of Claimed Subject Matter**

Independent claim 1 is directed to a method of executing a computer program in a computer device.

Claim 1 recites method of making secure the execution of a computer program (EXE) in a computing device (page 1, lines 4-66, Fig. 2) comprising a microprocessor (Fig. 2, element 10), the computer program (Fig. 2, SOURCE) including a set of instructions comprising a plurality of instructions (page 23, lines 2-6), which method comprises: said microprocessor executing (Fig. 2, element 10): a first step (E30), prior to the execution of the computer program, of calculating and storing a first signature (SIG1) representative of the intended execution of the set of instructions (Fig. 3, element E30, page 23, line 2 through page 25, line 8), a second step (E50), during the execution of the set of instructions (Fig. 3, element E40, page 25, lines 9-14), of calculating and storing a second signature (SIG2) representative of the execution of the set of instructions (Fig. 3, element E50, page 26, lines 4-18), and a step (E60) of detecting an anomaly in the execution of the set of instructions on the basis of the first signature (SIG1) and the second signature (SIG2) (Fig. 3, element E60, page 27, lines 9-15), wherein said set of instructions comprises at least one first instruction for initializing the calculation of the second signature (Appendices A-C, instructions A1, B3, C3, and D1), at least one second instruction depending upon the calculation mode

of the second signature (Appendices A-C, instructions and A12, B7, C7 and C8, and D11), and a third instruction, different than the at least one second instruction (Appendices A-C, instructions and A13, B8, C9 and C10, and D12), for comparing the second signature obtained according to the at least one second instruction with the first signature (Fig. 3, element 24, page 14, lines 8-34).

Independent claim 17 is directed to a computer device for executing a computer program.

Claim 17 recites computing device (Fig. 2) comprising a microprocessor for processing a computer program including a set of instructions (page 23, lines 2-6), comprising: said microprocessor executing: a means (12) for calculating and storing a first signature (SIG1) (Fig. 2, element 12), the first signature (SIG1) stored in a memory (Fig. 3, RAM) and the first signature (SIG1) is representative of the intended execution of the set of instructions prior to the execution thereof (page 28, lines 28-32), said set of instructions comprising at least one first instruction (Appendices A-C, instructions A1, B3, C3, and D1), for initializing the calculation of a second signature, at least one second instruction (Appendices A-C, instructions and A12, B7, C7 and C8, and D11) depending upon the calculation mode of the second signature, and a third instruction (Appendices A-C, instructions and A13, B8, C9 and C10, and D12), different than

the at least one second instruction, for comparing the second signature obtained according to the at least one second instruction with the first signature (page 38, lines 28 and 29).

Independent claim 26 is directed to a computer device for securely executing a computer program.

Claim 26 recites a computing device (Fig. 2) comprising a microprocessor (Fig. 2, element 10) for making secure the execution of a computer program (Fig. 2, SOURCE) including a set of instructions comprising a plurality of instructions (page 23, lines 2-6), which device comprises: a first register (REG1) (Fig. 2, REG1) for storing a first signature (SIG1) representative of the intended execution of the set of instructions (page 28, lines 28-32), means (22) (Fig. 3, element 22) for calculating and storing in said first register (REG1) (Fig. 3, REG1) or in a second storage register (REG2) (Fig. 3, REG2) during the execution of the set of instructions a second signature (SIG2) representative of the execution of the set of instructions (page 30, lines 12-16), and means (24) (Fig. 3, element 24) for detecting an anomaly in the execution of the set of instructions on the basis of the first signature (SIG1) and the second signature (SIG2) (page 31, line 25 through page 32, line 11), said set of instructions comprising at least one first instruction for initializing the calculation of the second signature (Appendices A-C, instructions A1, B3, C3, and D1), at

least one second instruction depending upon the calculation mode of the second signature (Appendices A-C, instructions and A12, B7, C7 and C8, and D11), and a third instruction, different than the at least one second instruction (Appendices A-C, instructions and A13, B8, C9 and C10, and D12), for comparing the second signature obtained according to the at least one second instruction with the first signature (page 31, lines 30-35).

(vi) **Grounds of Rejection to be Reviewed on Appeal**

The first issue on appeal is whether claims 1-4, 8, 9, 11-21, 23-28, 32, 33, and 35-40 are anticipated, in the meaning of 35 USC § 102(e), based on Naccache, U.S. Patent No. 7,168,065.

The second issue on appeal is whether claims 5-7, 29, and 30 are would have been obvious, in the meaning of 35 USC § 103(a), based on Naccache, U.S. Patent No. 7,168,065.

(vii) **Arguments**

(1) Arguments Concerning the First Ground of Rejection, Claims 1-4, 8, 9, 11-21, 23-28, 32, 33, and 35-40 would not have been anticipated based on Naccache.

Naccache aims at monitoring the execution of a program, that is to say the execution of each instruction of a set of instructions.

According to Naccache, a program that execution is to be monitored is executed in a program execution device (see Fig. 2, element 20) having a processor (see Fig. 2, element 4) linked to a monitoring unit (see Fig. 2, element 22) (that may be located in the program execution device or not). The program that execution is to be monitored comprises a set of instructions that are stored in an instruction register (see Fig. 2, element 2) of the program execution device.

The program that execution is to be monitored comprises n instructions (Inst.1 to Inst.n) to which two specific instructions (Inst.0 and Inst.n+1), called monitoring instructions in the following, are added.

The monitoring unit makes it possible to verify that each of the instructions Inst.1 to Inst.n has been loaded in the processor. To that end, such monitoring unit is functionally connected between the instruction register and the processor. Examples of the method implemented in the monitoring unit are given on Figs. 3 and 4.

The monitoring unit aims at monitoring the execution of the program that execution is to be monitored by detecting a first monitoring instruction in this program, by computing a signature according to the processed instructions of this program and by comparing the computed signature with a predetermined signature upon the detection of a second monitoring instruction in the program.

The computed signature is computed as a function of the value of the instructions loaded in the processor.

Accordingly, it is necessary to add two monitoring instructions in all the programs that execution is to be monitored by the disclosed monitoring unit. These specific instructions form the boundary of the set of instructions to monitor.

Therefore, the monitoring unit, if implemented as a monitoring program, should comprise a first detection instruction to detect the first monitoring instruction and a second detection instruction to detect the second monitoring instruction. The monitoring program comprises further instructions to compute signatures.

It is noted that the two monitoring instructions added to the programs that execution is monitored are directed to the monitoring unit only. They are not transmitted to the processor (see Naccache, col. 9, lines 30-31).

According to the teachings of Naccache, each instruction

of a program that execution is to be monitored is monitored by a monitoring unit and loaded in a processor. In other words, to be monitored, such a program must be executed in a program execution device that comprises the disclosed monitoring unit or that is connected to such a unit.

Moreover, due to the addition of the specific monitoring instructions, a program that is to be monitored according to the teachings of Naccache cannot be executed on a not specifically adapted program execution device that would not be able to process the added monitoring functions.

It is also observed that since program execution monitoring is performed in a monitoring unit, the monitoring characteristics do not primarily depend on the monitored program but on the monitoring unit's characteristics and/or settings.

Accordingly, when different programs are monitored in a same monitoring unit, they are monitored similarly except if the monitoring unit's characteristics and/or settings are changed.

The object of the claimed invention is directed to making secure the execution of a program. To that end, the program that execution is made secure is modified to add three different instructions, two that are related to the calculation and the comparison of a signature and one that is related to the calculation mode of this signature.

The added instructions are standard instructions and thus, they are executed like the other instructions of the

program that execution is made secure. In other words, the execution of a program that execution is made secure is performed on a standard device comprising a processor adapted to process a similar program that execution is not made secure. No monitoring or similar unit is required.

Accordingly, a program that execution is made secure according to the invention can be executed in standard devices.

Furthermore, since the instructions for controlling the execution of the program are in the program itself, it is possible to use different method of computing signatures for different programs executed on a single device without modifying any setting of the device. In other words, monitoring the execution of a program is done by the program itself.

In view of the preceding, it appears that the monitoring of the execution of a program is performed by a monitoring unit in Naccache while it is done by the program itself according to the claimed invention.

As a consequence, the meaning of the monitoring instructions is not the same for Naccache and for the claimed invention. Likewise, the way the monitoring instructions are added to the programs that execution is monitored and the way such monitoring instructions are handled or executed are not the same for Naccache and for the claimed invention.

Further, it is observed that Naccache is directed to a method and a device for monitoring the loading of instructions in

a processor. Indeed, the monitoring unit is connected between the instruction register and the processor so as to control the instructions transmitted to the processor. Accordingly, the execution of the instructions loaded in the processor is not, per se, monitored.

This is different as the claimed method and device for making secure the execution of a program wherein the signature used for determining whether or not the execution is correct is based on instruction execution results.

Naccache does not disclose a method and a device of making secure the execution of a computer program comprising three monitoring instructions and that the object of the claimed invention is different than the teachings of Naccache, it is shown that the program that execution is made secure in the meaning of the claimed invention cannot be assimilated to the monitoring program of Naccache.

The program that execution is made secure (the monitored program) is not the monitoring program

As observed above, Naccache discloses a method and a device for monitoring the execution of a program that is characterized by a set of instructions Inst.1 to Inst.n.

Such a method and a device are referred to as the monitoring unit. The instructions carried out by the monitoring unit, as shown on Figs. 3 and 4, i.e., the instructions of the monitoring program, are used to monitor the loading of the

instructions Inst.1 to Inst.n, i.e., the instruction of the monitored program, in a processor.

The instructions Inst.1 to Inst.n, stored in an instruction register of the program execution device, are not related to the instructions for carrying the steps of the algorithms described by reference to Figs. 3 and 4.

In other words, if the method shown on Figs. 3 and 4 are used to monitor the instructions of the program having the instruction Inst. 1 to Inst.n, the instructions of the method as shown on Figs. 3 and 4 are not monitored themselves. Indeed, nothing in Naccache relates to monitoring the execution of the steps depicted on Figs. 3 and 4.

Accordingly, a program implementing the flow diagram of Fig. 3 or 4 cannot be assimilated to a monitored program or a program that execution is made secure in the meaning of the claimed invention.

Therefore, contrarily to the opinion of the Examiner (See Office Action of August 11, 2010, Response to Arguments on page 2), the program that execution is made secure is not the monitoring program disclosed in Naccache and represented on Figs 3 and 4 but is the program comprising the instructions Inst.1 to Inst.n.

As previously stated, Naccache explicitly states that "the use of the monitoring unit requires the addition of *two new instructions* to the n instructions […] of the program

(see Naccache, col. 9, lines 18-20).

One of the instructions is used to initialize the calculation of a signature (see Naccache, col. 9, lines 27-28) in the monitoring unit while the other one is used to finalize the calculation of the signature and to compare the obtained value with a reference value (see Naccache, col. 9, lines 61-64) in the monitoring unit. These two new instructions are processed by the monitoring unit but they are not loaded in the processor.

Therefore, the program that execution is monitored, that is to say the program represented by the set of instructions Inst.1 to Inst.n memorized in the instruction register of the program execution device, comprises only two monitoring instructions that are specific.

This is different than the object of the claimed invention wherein three distinct standard instructions are used to control the execution of the program that execution is made secure.

As noted by the Examiner and according to the description, the computed signature is based upon a hash function (see Naccache, col. 10, lines 64-67). It is mentioned that, as a variant, other functions can be used, in particular a CRC function (see Naccache, col. 5, lines 53-58).

Since the signature is computed by the monitoring unit as a function of the processed instructions of the monitored program, such a function for computing the signature is set in

the monitoring program.

Therefore, Naccache discloses a monitoring program for calculating a signature as a function of the loading of the instructions of a program to be monitored in a processor where the signature is calculated according to a function determined in the monitoring program.

This is different as calculating a signature as a function of a calculation mode linked to an instruction of the program to be monitored.

As a consequence, not only Naccache does not disclose the use of an instruction in the program to be monitored for determining the function to be used for calculating a signature but such a step would have been meaningless in this context since monitoring the execution of a program is done in a monitoring unit and not directly by the execution of the monitored program itself.

Further, adding further monitoring instructions within the instructions of the program that execution is made secure would have been meaningless since such monitoring instructions would have been ignored by the disclosed monitoring unit. Indeed, the monitoring unit of Naccache is only looking for a first and a second monitoring instruction (see Naccache, Figs. 3 and 4), the other instructions being monitored instructions.

Claim 1

On page 4 of the Office Action, the Examiner asserts that Naccache col. 4, lines 35-36 disclose "calculating and storing a second signature (SIG2) representative of the execution of the set of instructions," as in claim 1.

As discussed above, Naccache is directed to a method and a device for monitoring the execution of a program by monitoring the loading of the instructions of this program in a processor. Accordingly, a signature is computed as a function of the value of the instructions loaded in a processor (see e.g., Naccache, Figs. 3 and 4, elements 40 and 49). The execution of the program is considered as being correct if the computed signature matches a predetermined value.

Therefore, Naccache does not disclose a step of calculating a signature representative of the execution of instructions. Indeed, even if an instruction is loaded in a processor, it is not necessarily executed by the processor since numerous events may occur between the instant an instruction is loaded and the instant the instruction is executed, if it is. Likewise, the processor may execute other instructions that would interfere with the instructions of the program to be monitored but that would not be taken into account for computing the signature used for monitoring the execution of the program.

As a consequence, Naccache does not disclose the instant feature of claim 1.

Further, in view of the implementation of the monitoring unit between the instruction register and the processor, the monitoring unit is not is position to compute a signature based on the execution of instructions. Accordingly, the step of "calculating and storing a second signature representative of the execution of the set of instructions" cannot be anticipated or rendered obvious from the teachings of Naccache.

For at least the reasons discussed above, claim 1 and the claims dependent therefrom are not anticipated by Naccache.

Claim 17

On page 9 of the Office Action, the Examiner asserts that Naccache col. 9, lines 25-30 and Fig. 3, element 34 discloses “said set of instructions comprising at least one first instruction for initializing the calculation of a second signature,” as in claim 17.

As discussed above, Naccache is directed to a method and a device for monitoring the execution of a program by monitoring the loading of the instructions of this program in a processor. Accordingly, a signature is computed as a function of the value of the instructions loaded in a processor (see e.g., Naccache, Figs. 3 and 4, elements 40 and 49). The execution of the program is considered as being correct if the computed signature matches a predetermined value.

Therefore, Naccache does not disclose a step of calculating a signature representative of the execution of instructions. Indeed, even if an instruction is loaded in a processor, it is not necessarily executed by the processor since numerous events may occur between the instant an instruction is loaded and the instant the instruction is executed, if it is. Likewise, the processor may execute other instructions that would interfere with the instructions of the program to be monitored but that would not be taken into account for computing the signature used for monitoring the execution of the program.

As a consequence, Naccache does not disclose the instant feature of claim 17.

Further, in view of the implementation of the monitoring unit between the instruction register and the processor, the monitoring unit is not is position to compute a signature based on the execution of instructions. Accordingly, the step of "said set of instructions comprising at least one first instruction for initializing the calculation of a second signature" cannot be deriving obviously from the teachings of Naccache.

Further on page 9 of the Office Action, it is asserted that Naccache, col. 9, lines 25-30, 34-40 and 51-55 disclose “said set of instructions comprising at least one first instruction for initializing the calculation of a second signature, at least one second instruction depending upon the

calculation mode of the second signature, and a third instruction, different than the at least one second instruction, for comparing the second signature obtained according to the at least one second instruction with the first signature" (emphasis added) as in claim 17.

Naccache explicitly states that "the use of the monitoring unit requires the addition of two new instructions to the n instructions … of the program" (col. 9, lines 18-20). One of the instructions is used to initialize the calculation of a signature (col. 9, lines 27-28) while the other one is used to finalize the calculation of the signature and to compare the obtained value with a reference value (col. 9, lines 61-64).

According to Naccache, the signature is based upon a hash function (col. 10, lines 64-67). However, it is also mentioned that, as a variant, other functions can be used, in particular a CRC function (col. 5, lines 53-58).

Thus, Naccache only discusses the use of two monitoring instructions in the computer program that execution is made secure (see Naccache col. 9, lines 18-19) and further instructions in the monitoring computer program (see Fig 3), the claimed invention is directed to three monitoring instructions in the computer program that execution is made secure.

Furthermore, according to Naccache, the two instructions added in the computer program that execution is made secure are used to identify the first and last instructions that

should be monitored, independently of the monitoring per se.

Accordingly, one of ordinary skill in the art would not be prompted to modify the teaching of Naccache to indicate, within the instructions of the computer program that execution is made secure, how the instructions should be monitored. Moreover, adding further monitoring instructions within the instructions of the computer program that execution is made secure would have been meaningless since such monitoring instructions would have been ignored by the disclosed monitoring method. Indeed, the monitoring method of Naccache is only looking for a first and a second monitoring instruction (see Figs. 3 and 4), the other instructions being monitored instructions. As a consequence, the invention as claimed cannot be anticipated or rendered obvious from Naccache.

For at least the reasons discussed above, claim 17 and the claims dependent therefrom are not anticipated by Naccache.

Claim 26

On page 5 of the Office Action, the Examiner asserts that Naccache col. 4, lines 35-36 disclose "means (22) for calculating and storing in said first register (REG1) or in a second storage register (REG2) during the execution of the set of instructions a second signature (SIG2) representative of the execution of the set of instructions," as in claim 26.

As discussed above, Naccache is directed to a method

and a device for monitoring the execution of a program by monitoring the loading of the instructions of this program in a processor. Accordingly, a signature is computed as a function of the value of the instructions loaded in a processor (see e.g., Naccache, Figs. 3 and 4, elements 40 and 49). The execution of the program is considered as being correct if the computed signature matches a predetermined value.

Therefore, Naccache does not disclose a step of calculating a signature representative of the execution of instructions. Indeed, even if an instruction is loaded in a processor, it is not necessarily executed by the processor since numerous events may occur between the instant an instruction is loaded and the instant the instruction is executed, if it is. Likewise, the processor may execute other instructions that would interfere with the instructions of the program to be monitored but that would not be taken into account for computing the signature used for monitoring the execution of the program.

As a consequence, Naccache does not disclose the instant feature of claim 26.

Further, in view of the implementation of the monitoring unit between the instruction register and the processor, the monitoring unit is not is position to compute a signature based on the execution of instructions. Accordingly, the step of "means (22) for calculating and storing in said first register (REG1) or in a second storage register (REG2) during the

execution of the set of instructions ***a second signature (SIG2) representative of the execution of the set of instructions***" (emphasis added) cannot be anticipated or rendered obvious from the teachings of Naccache.

For at least the reasons discussed above, claim 26 and the claims dependent therefrom are not anticipated by Naccache.

(2) Arguments Concerning the Second Ground of Rejection, Claims 5-7, 29, and 30 would not have been obvious based on Naccache.

Claims 5-7, 29, and 30 are allowable as being dependent on otherwise allowable base claims as argued above.

Therefore, for at least the reasons discussed above, Naccache fails to render obvious the features of claims 5-7, 29, and 30.

Conclusion

Appellants respectfully urge that the rejections on appeal should not be maintained, and respectfully requests that these rejections be reversed.

The fee for the Appeal Brief in the amount of $540.00 is being paid online herewith by credit card.

If necessary, the Commissioner is hereby authorized in this, concurrent, and future submissions, to charge any underpayment or credit any overpayment to Deposit Account No. 25-0120 for any additional fees required under 37 C.F.R. § 1.16 or under 37 C.F.R. § 1.17.

Respectfully submitted,

YOUNG & THOMPSON

/James J. Livingston, Jr./

James J. Livingston, Jr.
Reg. No. 55,394
209 Madison Street, Suite 500
Alexandria, VA 22314
Telephone (703) 521-2297
Telefax (703) 685-0573
(703) 979-4709

JJL/jr

January 18, 2011
(timely filed because the USPTO was closed on January 17, 2011)

Enclosure: Claims Appendix

(viii) **Claims Appendix**

1. A method of making secure the execution of a computer program (EXE) in a computing device comprising a microprocessor, the computer program including a set of instructions comprising a plurality of instructions, which method comprises:

said microprocessor executing:

- a first step (E30), prior to the execution of the computer program, of calculating and storing a first signature (SIG1) representative of the intended execution of the set of instructions,

- a second step (E50), during the execution of the set of instructions, of calculating and storing a second signature (SIG2) representative of the execution of the set of instructions, and

- a step (E60) of detecting an anomaly in the execution of the set of instructions on the basis of the first signature (SIG1) and the second signature (SIG2),

wherein said set of instructions comprises at least one first instruction for initializing the calculation of the second signature, at least one second instruction depending upon the calculation mode of the second signature, and a third instruction, different than the at least one second instruction, for comparing the second signature obtained according to the at least one second instruction with the first signature.

2. The method according to claim 1, wherein the first calculation and storage step (E30) is executed during the generation of the instructions (A1, A13) of the computer program.

3. The method according to claim 1, wherein the second signature (SIG2) stored during the second calculation and storage step (E50) is retained in memory during the execution of at least one second instruction following the set of instructions.

4. The method according to claim 1, wherein:

- the first signature (SIG1) is obtained from the number of instructions in the set of instructions,

- the second signature (SIG2) is obtained from the number of instructions from the set of instructions that have been executed, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

5. The method according to claim 1, wherein:

- the first signature (SIG1) is obtained from the number of instructions in the set of instructions,

- the second signature (SIG2) is obtained from the number of instructions from the set of instructions that have not been executed, this second signature (SIG2) being calculated from

the first signature (SIG1), and in that

- the detection step (E60) detects an execution anomaly when the value of the second signature (SIG2) is not zero after the execution of the set of instructions.

6. The method according to claim 5, wherein an interrupt of the computer program is triggered when the value of the second signature (SIG2) is below a predetermined threshold.

7. The method according to claim 5, wherein the first signature (SIG1) and the second signature (SIG2) are retained in memory during the execution of the program in the same register (REG1).

8. The method according to claim 1, wherein:

- the first signature (SIG1) is obtained from the code of a critical instruction of the set of instructions,

- the second signature is obtained from the code of the critical instruction, that code being stored at the same time as or after the execution of the critical instruction, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

9. The method according to claim 1, wherein:

- the first signature (SIG1) is obtained from the address of a critical instruction of the set of instructions, the address being obtained during or after the generation of the executable code of the set of instructions,

- the second signature (SIG2) is obtained from the address of the critical instruction, that address being stored (E30) at the same time as or after the execution (E30) of the critical instruction, and

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

10. (cancelled)

11. The method according to claim 1, wherein:

- the first signature (SIG1) and the second signature (SIG2) are error detector codes (CRC1, CRC2) calculated from the code or from an address of an instruction of the set of instructions, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

12. The method according to claim 11, wherein the error detector codes (CRC1, CRC2) are cyclic redundancy check codes.

13. The method according to claim 11, wherein the error detector codes are obtained by the logical combination (XOR) of the code or an address of at least one instruction of the set of instructions.

14. The method according to claim 1, wherein:

- the first signature (SIG1) and the second signature (SIG2) are respectively obtained during the generation and t he execution of the instructions from at least two elements chosen from:

. the number of instructions in the set of instructions,

. the code of at least one instruction of the set of instructions,

. the address of at least one instruction of the set of instructions, and

. an error detector code calculated from the code or an address of at least one critical instruction of the set of instructions, the address being obtained during or after the generation of the executable code of the set of instructions, and in that

- the detection step (E60) detects an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

15. The method according to claim 1, wherein it includes a step (E70) of destroying at least a portion of the system on which the computer program is executed, this step of destroying being made when an execution anomaly is detected in the detection step.

16. The method according to claim 1, wherein the first signature (SIG1) is generated automatically (E30).

17. A computing device comprising a microprocessor for processing a computer program including a set of instructions, comprising:

said microprocessor executing:

a means (12) for calculating and storing a first signature (SIG1), the first signature (SIG1) stored in a memory and the first signature (SIG1) is representative of the intended execution of the set of instructions prior to the execution thereof, said set of instructions comprising at least one first instruction for initializing the calculation of a second signature, at least one second instruction depending upon the

calculation mode of the second signature, and a third instruction, different than the at least one second instruction, for comparing the second signature obtained according to the at least one second instruction with the first signature.

18. The computing device according to claim 17, wherein the means (12) for calculating and storing the first signature (SIG1) are adapted to calculate and store information obtained from the number of instructions of the set of instructions.

19. The computing device according to claim 17, wherein the means (12) for calculating and storing the first signature (SIG1) are adapted to obtain and store information obtained from the code of a critical instruction of the set of instructions.

20. The computing device according to claim 17, further comprising:

means (14) for generating executable code from the computer program (SOURCE).

21. The computing device according to claim 20, wherein the means for calculating and storing the first signature (SIG1) are adapted to obtain and store information obtained from the address of a critical instruction, the information being obtained

of the set of instructions by the means (14) for generating executable code.

22. (cancelled)

23. The computing device according to claim 17, wherein the means (12) for calculating and storing the first signature (SIG1) are adapted to calculate and store information obtained from an error detector code (CRC1) calculated from the code or an address of at least one instruction of the set of instructions.

24. The computing device according to claim 23, wherein the error detector code (CRC1) is a cyclic redundancy check code.

25. The computing device according to claim 23, wherein the error detector code is obtained by a logical combination (XOR) of the code or an address of at least one instruction of the set of instructions.

26. A computing device comprising a microprocessor for making secure the execution of a computer program including a set of instructions comprising a plurality of instructions, which device comprises:

- a first register (REG1) for storing a first signature (SIG1) representative of the intended execution of the set of instructions,

- means (22) for calculating and storing in said first register (REG1) or in a second storage register (REG2) during the execution of the set of instructions a second signature (SIG2) representative of the execution of the set of instructions, and

- means (24) for detecting an anomaly in the execution of the set of instructions on the basis of the first signature (SIG1) and the second signature (SIG2),

said set of instructions comprising at least one first instruction for initializing the calculation of the second signature, at least one second instruction depending upon the calculation mode of the second signature, and a third instruction, different than the at least one second instruction, for comparing the second signature obtained according to the at least one second instruction with the first signature.

27. The computing device according to claim 26, wherein the calculation and storage means are adapted to retain the second signature (SIG2) in the second register (REG2) during the execution of at least one second instruction following the set of instructions.

28. The computing device according to claim 26, wherein, the first signature (SIG1) being obtained from the number of instructions of the set of instructions, the second

signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the number of instructions of the set of instructions that have been executed and in that the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

29. The computing device according to claim 26, wherein, the first signature (SIG1) being obtained from the number of instructions of the set of instructions, the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the number of instructions of the set of instructions that have not been executed, this second signature (SIG2) being calculated from the first signature (SIG1), and in that detection means (24) detect an execution anomaly when the value of second signature (SIG2) is not zero after the execution of the set of instructions.

30. The computing device according to claim 29, wherein it further includes means for triggering an interrupt of the computer program when the value of the second signature (SIG2) is below a predetermined threshold.

31. (cancelled)

32. The computing device according to claim 26, wherein, the first signature (SIG1) being obtained from the code of a critical instruction of the set of instructions, the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the code of the critical instruction, the code being stored at the same time as or after the execution of the critical instruction, and in that the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

33. The computing device according to claim 26, wherein, the first signature (SIG1) being obtained from the address of a critical instruction of the set of instructions, the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained from the address of the critical instruction, that address being stored at the same time as or after the execution of the critical instruction, and in that the detection means detect an execution anomaly when the first and second signatures are different after the execution of the set of instructions.

34. (cancelled)

35. The computing device according to claim 26, wherein, the first signature (SIG1) and the second signature (SIG2) being error detector codes (CRC1, CRC2) calculated from the code of an instruction of the set of instructions, the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

36. The computing device according to claim 35, wherein the error detector codes (CRC1, CRC2) are cyclic redundancy check codes.

37. The computing device according to claim 35, wherein the error detector codes are obtained by a logical combination (XOR) of the code or an address of at least one instruction of the set of instructions.

38. The computing device according to claim 26, wherein, the first signature (SIG1) being obtained from at least two elements chosen from:

- the number of instructions of the set of instructions,

- the code of at least one instruction of the set of instructions,

- the address of at least one instruction of the set of instructions, and

- an error detector code calculated from the code or the address of at least one instruction of the set of instructions,

the second signature (SIG2) calculated and stored by the calculation and storage means (22) is obtained in a similar manner from the at least two elements during the execution of the instructions and in that the detection means (24) detect an execution anomaly when the first signature (SIG1) and the second signature (SIG2) are different after the execution of the set of instructions.

39. The computing device according to claim 26, wherein it further includes means for destroying at least a portion of the computer program.

40. A microcircuit card characterized in that it includes a securing device (100) according to claim 26.

(ix) **Evidence Appendix**

None.

(x) **Related Proceedings Appendix**

None.